AN EMPIRICAL STUDY WITH OBJECT-RELATIONAL DATABASES METRICS

Mario Piattini[1], Coral Calero[1], Houari Sahraoui[2] , Hakim Lounis [3]
[1]E.S. Informática
University of Castilla-La Mancha
Ciudad Real (Spain)
e-mail: {mpiattin, ccalero}@inf-cr.uclm.es
[2]Dep.d'Informatique et Recherche Opérationelle
Université de Montréal
Montréal (Canada)
e-mail: sahraouh@iro.umontreal.ca
[3]Centre de Recherche Informatique de Montreal
550 Sherbrooke west #100
Montréal QC H3A 1B9 Canada
+1 514 840 1234
hlounis@crim.ca

Abstract

It is important that the software products, and obviously databases, are evaluated for every relevant quality characteristic, using validated or widely accepted metrics. It is also important to validate the metrics from a formal point of view in order to ensure its usefulness. However, into the aspects of software measurement, the research is needed, from theoretical but also from a practical point of view. So, it is necessary to do experiments to validate the metrics. We have developed an experiment in order to validate some object-relational databases metrics. The experiment was done on CRIM (Centre de Recherche Informatique de Montréal) from Canada and repeated on the University of Castilla-La Mancha from Spain. The results obtained on both experiments were very similar, so we can conclude that both, metric TS (Table Size) and DRT (Depth of Referential Tree) seem to be good indicators for the maintainability of a table.

1. Introduction

It is important that the software products, and obviously databases, are evaluated for every relevant quality characteristic, using validated or widely accepted metrics. These metrics could help designers, to choose the most maintainable, among semantically equivalent alternative schemata. Because of this, we think it is very important to measure databases and to understand their contribution to the overall IS maintainability.

We have put forward different measures (for internal attributes) in order to measure the complexity that affects the maintainability (an external attribute) of the object-relational databases which is useful for control its quality.

It is also important to validate the metrics from a formal point of view in order to ensure its usefulness. Several frameworks for measure characterization have been proposed. Some of them (Briand et al., 1996; Weyuker, 1988; Briand and Morasca, 1997) are based on axiomatic approaches. The goal of these approaches is merely definitional by proposing formally desirable properties for measures for a given software attribute, so axioms must be used as guidelines for the definition of a measure. Others (Zuse, 1998) are based on measurement theory which specifies the general framework in which measures should be defined. Some of the presented metrics have been formalized from both points of view, axiomatic approach (Piattini et al., 1998) and measurement theory (Calero et al., 1999).

However, into the aspects of software measurement, the research is needed (Neil, 1994), from theoretical but also from a practical point of view (Glass, 1996). So, it is necessary to do experiments to validate the metrics. Empirical validation can be used to investigate the association between proposed software metrics and other indicators of software quality as maintainability (Harrison et al., 1998)

In this line we developed an experiment in order to validate some object-relational database metrics (presented on the workshop which took place on the past ECOOP edition in Lisbon). First, the experiment was done on CRIM (Centre de Recherche Informatique de Montreal) from Canada and we obtained the first conclusions about our metrics.

In order to make sure that the results of the experiment were minimally conclusive we replicated it. So, we repeated the same experiment with people from the University of Castilla-La Mancha from Spain. The results obtained on the replication were very similar to the obtained the first time, so some of the conclusions seem to be important.

In this paper we present on section 2 the metrics with which we have worked. In section 3 we show both experiments and the results obtained for each one and the conclusions we can draw out from both. Conclusions and future work come on the last section.

2. Metrics for object-relational databases

In this section we present the metrics with which we have worked on the experiment. These metrics are the same as the presented on the workshop on the past ECOOP edition. The metrics complete definition and an example can be found on Calero et al. (1999).

In general, an object-relational database schema is composed by a number of related tables. Every table has columns that can be defined as a simple data or as a complex data. The type of a simple data may be one of the classic data types as integer, number or character. A complex data is defined above a class (or a UDT, user defined type), which can be related with other classes (types) by generalization or inheritance associations. For this kind of database we can propose table related metrics (when we apply the metrics to a table) and schema oriented metrics (when the metrics are applied to the schema). As table oriented metrics we propose, being T a table, the metrics TS, DRT(T), RD(T), PCC(T), NIC(T), NSC(T) defined as follows:

TS metric. We define the table size (TS) as the sum of the total size of the simple columns (TSSC) and the total size of the complex columns (TSCC) in the table.

DRT(T) metric. Depth of Relational Tree of a table T (DRT(T)) is defined as the longest referential path between tables, from the table T to any other table in the schema database

RD(T) metric. Referential Degree of a table T (RD(T)) is defined as the number of foreign keys in the table T.

PCC(T) metric. Percentage of complex columns of a table T.

NIC(T) metric. Number of involved classes. This measures the number of all classes that compose the types of the complex columns of T using the generalization and the aggregation relationships.

NSC(T) metric. Number of shared classes. This measures the number of involved classes for T that are used by other tables.

At the schema level, we can apply the next metrics:

DRT metric. Depth of referential Tree, defined of the longest referential path between tables in the database schema.

RD metric. Referential Degree is defined as the number of foreign keys in the schema database.

PCC metric. Percentage of complex columns in the schema database.

NIC metric. Number of involved classes, number of all classes that composes the types of the complex columns, using the generalization and aggregation relationships, of all tables in the schema.

NSC metric. Number of shared classes, number of shared classes by tables of the schema.

3. Experiments

In this section, we present the experiment developed in order to evaluate whether the proposed measures can be used as indicators for estimating the maintainability of an OR database.

Data Collection

Five object-relational databases were used in this experiment with the average of 10 relations per database (ranging from 6 to 13). These databases were originally relational ones. For the purpose of the experiment, they were redesigned as OR databases. A brief description of these databases is given in table 1.

Database	Number of tables	Average attributes/table	Average complex attributes/table
Airlines	6	4,16	1,83
Animals	10	2,7	0,6
Library	12	2,91	0,75
Movies	9	4,33	0,88
Treebase	13	3,46	0,86

Table 1. Databases used in he experiment

Five people participated in the experiment the first time we made it (Canadian experiment): one researcher, two research assistants and two graduate students. All of them are experienced in both relational databases and object-oriented programming. On the first experiment, one person did not complete the experiment, and we had to discard his partial results. So, in the replication (Spanish experiment) only four people made the experiment. Also all of them are experienced in both relational databases and object-oriented programming

The people were given a form, which include for each table, a triplet of values to compute using the corresponding schema. These values are those of three measures TS, DRT and RD. Our idea is that to compute these measures, we need to understand the subschema (objects and relations) defined by the concerned table. A table (and then the corresponding subschema) is easy to understand if (almost) all the people find the right values of hte metrics in a limited time (2 minutes per table). We wanted to measure understandability, we decided to give our people a limited time to finish the tests they had carry out and then, use all the tests that had been answered in the given time and in a correct way (following all the indications given for the development of the experiment). So, our study would focus on the amount of metrics correctly calculated. Formally, a value 1 is assigned to the maintainability of a table if at least 10 of 12 measures are computed correctly in the specified time (4 people and 3 measures). A value 0 is assigned otherwise. The tables are given to the people in a random order and not by database.

Validation Technique

To analyze the usefulness of the metrics proposed, we used two techniques: C4.5 (Quinlan, 1993), a machine learning algorithms and RoC (Ramoni and Sebastiani, 1999), a robust Bayesian classifier.

C4.5 belongs to the divide and conquer algorithms family. In this family, the induced knowledge is generally represented by a decision tree. The principle of this approach could be summarized by this algorithm:

If the examples are all of the same class
Then - create a leaf labelled by the class name;
Else - select a test based on one attribute;
- divide the training set into subsets, each associated to one of the possible values of the tested attribute;
- apply the same procedure to each subset;
Endif.

The key step of the algorithm above is the selection of the "best" attribute to obtain compact trees with high predictive accuracy. Information theory-based heuristics have provided effective guidance for this division process. C4.5 induces Classification Models, also called Decision Trees, from data. It works with a set of examples where each example has the same structure, consisting of a number of attribute/value pairs. One of these attributes represents the class of the example. The problem is to determine a decision tree that correctly predicts the value of the class attribute (i.e., the dependent variable), based on answers to questions about the non-class attributes (i.e., the independent variables). In our study, the C4.5 algorithm partitions continuous attributes (the database metrics), finding the best threshold among the set of training cases to classify them on the dependent variable (i.e. understandability of the database schemes).

RoC is a Bayesian classifier. It is trained by estimating the conditional probability distributions of each attribute, given the class label. The classification of a case, represented by a set of values for each attribute, is accomplished by computing the posterior probability of each class label, given the attributes values, by using Bayes' theorem. The case is then assigned to the class with the highest posterior probability.

The simplifying assumptions underpinning the Bayesian classifier are that the classes are mutually exclusive and exhaustive and that the attributes are conditionally independent once the class is known. RoC extends the capabilities of the Bayesian classifier to situations in which the database reports some entries as unknown. It can then train a Bayesian classifier from an incomp lete database.

One of the great advantages of C4.5 comparing to RoC is that, it produces a set of rules, directly understandable by software manager and engineers.

Results

As specified in validation technique section, we applied RoC and C4.5 to evaluate the usefulness of the OR metrics in estimating the maintainability of the tables in an OR schema.

- RoC technique

Using the cross-validation technique, the algorithm RoC was applied 10 times on the 50 examples obtained from the 50 tables of the five schematas (500 cases). 369 cases were correctly estimated for the Canadian experiment (accuracy 73.8%) and 407 cases for the Spanish one (accuracy 81.4%) and all the other cases in both experiments were missclassified. Contrary to C4.5, RoC does not propose a default classification rule which guaranteed a coverage of all the proposed cases. However, in this experiment, it succeeded to cover all the 500 cases (coverage of 100%). These results are summarized in the table 2.

	Spain	**Canada**
Correct:	407	369
Incorrect:	93	131
Not classified:	0	0
Accuracy:	81.4 %	73.8 %
Coverage:	100.0 %	100.0 %

Table 2. RoC quantitative results with data from Spain and from Canada

RoC produces the model presented in figure 1 with the Canadian data. From this model, it is hard to say which metric is more relevant than another in an absolute manner. However, we can notice that when TS is smaller, the probability that the table is understandable is higher (for example 55% for TS <= 3). This probability decrease when the table size increase (9.5% for TS >10). Inversely, the same probability increase in estimating the table that are not understandable (varying from 13.6% for TS <= 3 to 33.6% for TS >10). For DRT and RD, it is hard to draw a conclusion since no uniform variation is shown. This can be explained by the fact that for the sample used in this experiment, the values of DRT and RD are in defined in a narrow range ([0, 3] and [0, 5]).

Canadian Model

0 (0.547 0.547)
1 (0.453 0.453)
TS

	(1 . 3)	**(3 . 5)**	**(5 . 10)**	**(10 . 17.5)**
0	(0.136 0.136)	(0.193 0.193)	(0.336 0.336)	(0.336 0.336)
1	(0.543 0.543)	(0.233 0.233)	(0.129 0.129)	(0.095 0.095)

DRT

	0	**1**	**2**	**3**
0	(0.336 0.336)	(0.221 0.221)	(0.193 0.193)	(0.250 0.250)
1	(0.336 0.336)	(0.371 0.371)	(0.233 0.233)	(0.060 0.060)

RD

	0	**1**	**2**	**3**	**4**	**5**
0	(0.319 0.319)	(0.319 0.319)	(0.148 0.148)	(0.09 0.09)	(0.062 0.062)	(0.062 0.062)
1	(0.316 0.316)	(0.247 0.247)	(0.316 0.316)	(0.040 0.040)	(0.040 0.040)	(0.040 0.040)

Figure 1. The model generated by RoC with data from Canada

RoC produces the model presented in figure 2 with the Spanish data. The conclusions from this second model are the same as the first one because the models are very similar.

Spanish Model

0 (0.453 0.453)
1 (0.547 0.547)
TS

	(1 . 3)	**(3 . 5)**	**(5 . 10)**	**(10 . 17.5)**
0	(0.095 0.095)	(0.129 0.129)	(0.405 0.405)	(0.371 0.371)
1	(0.507 0.507)	(0.279 0.279)	(0.107 0.107)	(0.107 0.107)

DRT

	0	**1**	**2**	**3**
0	(0.371 0.371)	(0.198 0.198)	(0.164 0.164)	(0.267 0.267)
1	(0.307 0.307)	(0.364 0.364)	(0.250 0.250)	(0.079 0.079)

RD

	0	**1**	**2**	**3**	**4**	**5**
0	(0.351 0.351)	(0.316 0.316)	(0.075 0.075)	(0.109 0.109)	(0.075 0.075)	(0.075 0.075)
1	(0.290 0.290)	(0.262 0.262)	(0.348 0.348)	(0.033 0.033)	(0.033 0.033)	(0.033 0.033)

Figure 2. The model generated by RoC with data from Spain

- C4.5 technique

The results obtained for the Canadian experiment are shown in table 3. The model of C4.5 was very accurate in estimating the maintainability of a table, 94% and resent a high level of completeness (up to 100% for not understandable tables) and correctness (up to 100% for understandable tables).

		Predicted maintainability		
		0	1	Completeness
Real	0	28	0	100%
Maintainability	1	3	19	86.36%
	Correctness	90.32%	100%	

Accuracy = 94%

Table 3. C4.5 quantitative results from the Canadian experiment

And the rules obtained with C4.5 are:

```
Rule 1:
TS <= 9 ∧ DRT = 0 ∧ NSC = 0 -> class 1                [84.1%]
Rule 2:
TS <= 3 ∧ RD > 1 -> class 1                           [82.0%]
Rule 7:
TS <= 9 ∧ DRT <= 2 ∧ NIC > 0 ∧ NSC = 0 -> class 1     [82.0%]
Rule 5:
TS > 9 -> class 0                                     [82.2%]
Rule 6:
DRT > 2 -> class 0                                    [82.0%]

Default class: 0
```

Figure 3. C4.5 estimation model from the Canadian data

TS seems to be an important indicator for the maintainability of the tables. Rules 1, 2 and 7, which determine if a table is maintainable, have all as part of the conditions that TS must be small. Inversely, in rule 5, it is stated a large size is sufficient to declare the table as not understandable. A small DRT is also required for rules 1 and 7 as partial condition to classify the table as understandable. In the same time, a high value of DRT means that the table is hard to understand (rule 6). RD does not represent an interesting indicator.

The results obtained for the Spanish experiments are shown in table 4. In this case the accuracy in estimating the maintainability was 94% and the levels of completeness and correctness were smaller than the Canadian experiment but were also very high.

		Predicted maintainabilty		
		0	1	Completeness
Real	0	21	1	95.45%
Maintainabilty	1	3	25	89.29%
	Correctness	87.5%	96.15%	

Accuracy = 92%

Table 4. C4.5 quantitative results from the Spanish experiment

And the rules obtained with C4.5 are:

Rule 1:
TS <= 5 ∧ DRT <= 2 -> class 1 [89.4%]
Rule 3:
TS > 5 ∧ PCC <= 66 -> class 0 [82.3%]
Rule 2:
DRT > 2 -> class 0 [66.2%]
Default class: 1

Figure 4. C4.5 estimation model from the Spanish data

The model rules is smaller than the one of the first experiment but it confirms that (at least for the studied sample) TS and DRT are good indicators and not RD.

Both experiments and both techniques find out that the table size metric (TS) is a good indicator for the maintainability of a table. The depth of the referential tree metric (DRT) is also presented as an indicator by C4.5 on both experiments and the referential degree metric (RD) does not seem to have a real impact on the maintainability of a table.

4. Conclusions

We have presented a first approach for measuring object-relational databases using three different metrics. To validate our measures for the maintainability purpose, we have used 5 existing relational databases redesigned accordingly to the OR model. We have applied two deferent techniques: C4.5, a machine learning algorithm and RoC, a Bayesian theorem-based algorithm.

The results of our experimentation demonstrate that our measures can estimate with a higher level of accuracy the maintainability of OR tables. Two estimation models have been generated according to the two techniques from both Canadian and Spanish experiments. We have found that a sub-set of our measures proved to be quite accurate (table size and depth of referential tree). This suggests that these measures can be reasonably used as indicators for the maintainability of a table.

Moreover, controlled experiments have problems (like the large number of variables that causes differences, deal with low level issues, microcosm of reality and small set of variables) and limits (do not scale up, are done in a class in training situations, are made in vitro and face a variety of threats of validity). Then, is convenient to run multiple studies, mixing controlled experiments and case studies. We are now working on this last kind of empirical validation with our metrics.

References

- Briand, L.C., Morasca, S. and Basili, V. (1996). Property-based software engineering measurement. IEEE Transactions on Software Engineering, Vol.22(1). pp. 68-85.
- Briand L.C. and Morasca S. (1997). Towards a Theoretical Framework for Measuring Software Attributes. *Proceeding of the Fourth International, Software Metrics Symposium*, 119-126
- Calero, C., Piattini, M., Ruiz, F. and Polo, M. (1999), Validation of metrics for Object-Relational Databases, *International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (ECOOP99),* (Lisbon ,Portugal. June 1999), 14-18

- Glass, R. (1996). The Relationship Between Theory and Practice in Software Engineering. IEEE Software, November, Vol.39 (11). pp 11-13.
- Harrison, R., Counsell, S. And Nithi, R. (1998), Coupling metrics for Object-Oriented Design, *5th. International Symposium on Software Metrics*, IEEE Computer Society. Bethesda, Maryland, 20-21 November.
- Neil, M. (1994) Measurement as an Alternative to Bureaucracy for the Achievement of Software Quality. Software Quality Journal Vol.3 (2). pp. 65-78.
- Piattini, M., Calero, C., Polo, M. and Ruiz, F. (1998). Maintainability in Object-Relational Databases. . Proc of The European Software Measurement Conference FESMA 98, Antwerp, May 6-8, Coombes, Van Huysduynen and Peeters (eds.). pp. 223-230.
- Quinlan, J.R., (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers.
- Ramoni, M. and Sebastiani, P. (1999)Bayesian methods for intelligent data analysis. In M. Berthold and D.J. Hand, editors, *An Introduction to Intelligent Data Analysis*, New York,. Springer.
- Weyuker, E.J. (1988). Evaluating software complexity measures. IEEE Transactions on Software Engineering Vol.14(9). pp. 1357-1365.
- Zuse, H. (1998). A Framework of Software Measurement. Berlin, Walter de Gruyter.